



Universidade de Brasília

Faculdade de Tecnologia da Universidade de Brasília (FT/UnB)

Departamento de Engenharia Elétrica

Sistemas Microprocessados (ENE0056)

Projeto Final

Vaso de Planta IOT

João Tito Silva

180123301@aluno.unb.br

Ronald Cesar Dias de Oliveira

180047205@aluno.unb.br

Prof. Eduardo Peixoto

Turma A

Brasília, 06 de Maio de 2022

Índice

1. Objetivos	2
2. Introdução	2
3. Implementação	2
3.1. GPIO	2
3.2. Timers	2
3.3. SPI	2
3.4. ADC	2
3.5. Módulo Wifi ESP286	2
3.6. Bomba D'água	3
3.7. Funcionamento do Sistema	3
4. Resultados	3

1. Objetivos

O projeto teve como objetivo a implementação de um sistema de irrigação e medição controlado pelo MSP430 através de WiFi por meio do módulo ESP286 NodeMCU ESP-12.

2. Introdução

Visto que o projeto foi desenvolvido em dupla, uma forma de modularizar a implementação foi utilizar o repositório de códigos GitHub que utiliza o controlador de versões Git para guardar o histórico da modificação dos arquivos conforme o desenvolvimento. O código está disponível no repositório. Dessa forma, algumas bibliotecas foram implementadas e testadas individualmente para facilitar a integração do sistema como um todo.

3. Implementação

3.1. GPIO

Algumas Macros foram definidas para facilitar o uso dos componentes internos ao MSP como as chaves e os leds. A fim de configurar os pinos do MSP430 a biblioteca de GPIO configura os modos input, output, inPullUp, inPullDown e module.

As funções dessa biblioteca tem as funções de configurar, ler, escrever, alternar e por fim uma inicialização dos botões e leds.

3.2. Timers

A partir da biblioteca de timers que foi implementada é possível configurar as unidades de tempo desejadas entre microsegundos a horas podendo utilizar os clocks aclk e smclk, além de uma configuração para gerar o trigger de PWM com carga de 50% por meio do timer A0.

Algumas possibilidades são a de instanciar um novo timer, verificar se está contando, configurar o TA0 para um PWM ou então gerar um debounce por meio da trava de software esperando a flag do timer.

3.3. SPI

Macros para mosi, miso, metade do sclk, e um período de clock de 10 ms. Configuração do SPI no UCB0, função de transferência que retorna o byte recebido e uma configuração de teste para Slave no UCB1.

3.4. ADC

Vetor de inteiros que é usado para armazenar os resultado da conversões A/D por meio de interrupção do ADC e a própria configuração do ADC no modo 12 bits, utilizando a sequência de multi-channel sem repetição, o TA0.1 como fonte de trigger e o SMCLK como clock de amostragem.

3.5. Módulo Wifi ESP286

O módulo ESP286 NodeMCU ESP-12 foi configurado para gerar um canal TCP e foi configurado para enviar e receber bytes por meio do SPI, para isso foi utilizada a configuração de Master, com polaridade baixa e fase alta, full duplex e o clock foi dividido por 400 por ser consideravelmente mais alto do que o do MSP430.

3.6. Bomba D'água

A bomba de água utilizada funciona na faixa de tensão de 3V a 5V, a alimentação foi realizada por meio da conexão com o circuito integrado L293D que utilizou a alimentação de 5V do MSP430 e um pino de GPIO responsável pela instrução de irrigação.

3.7. Funcionamento do Sistema

Utilizando as configurações implementadas em SPI, o MSP430 foi configurado para atuar como escravo, o NodeMCU como mestre e os comandos foram implementados no próprio módulo. Ao conectar um celular por meio de um aplicativo que acesse por meio de um protocolo TCP é possível enviar comandos para o NodeMCU que envia e recebe os bytes através dos canais de comunicação do próprio SPI, dessa forma é possível medir temperatura, luminosidade, umidade e ativar a bomba d'água.

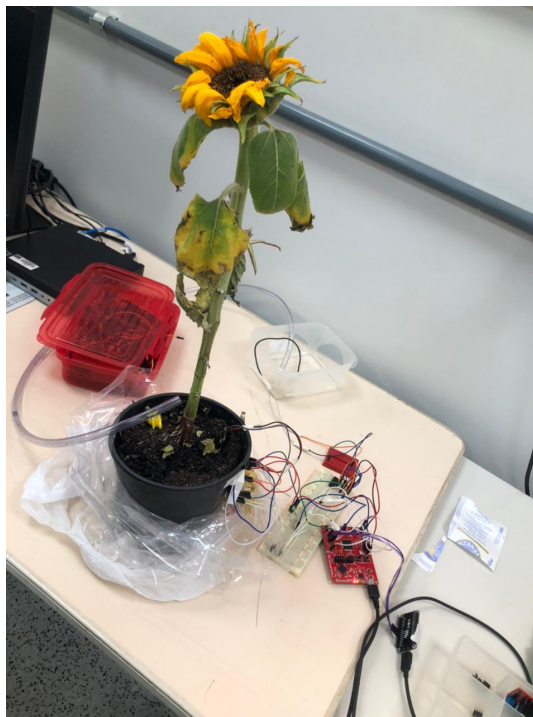


Figura 1: Sistema de Irrigação e Medição

4. Resultados

O vaso de plantas funcionou com duas instruções enviadas por TCP para o NodeMCU, como esperado, as quais foram um comando para medir leituras de sensores e outro para irrigar a bomba. A comunicação SPI funcionou corretamente após serem utilizadas as configurações corretas.

O comando de medir funcionou adequadamente, lendo os sensores de temperatura (LM335), o sensor de luminosidade (LDR) e de umidade do solo. No entanto, devido à baixa quantidade de medições realizadas no modo de sequência de canais sem repetição, as medidas de temperatura e luminosidade apresentaram certo ruído. O comando de irrigar corretamente ativou a bomba d'água por 5 segundos, como esperado.

Algumas melhorias que poderiam ser feitas no projeto seriam utilizar o protocolo HTTP com páginas em HTML para facilitar o uso da interface com o usuário, a utilização do modo de sequência de canais com repetição no ADC para reduzir os ruídos e o uso de uma fonte externa com maior capacidade de corrente para alimentar a bomba d'água, tal como pilhas externas. Também seria possível utilizar o low power mode no MSP.

Referências bibliográficas

- [1] MSP430. Userguide. [Online]. Available: https://www.ti.com/lit/ug/slau208q/slau208q.pdf?ts=1651810499164&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430F5659
- [2] E. N. 12-ESP. Datasheet. [Online]. Available: https://www.electrodragon.com/w/File:Schematic_esp-12-q.png
- [3] LM335. Circuito arduino. [Online]. Available: <https://modelect.wordpress.com/2012/07/12/lm335-measuring-temperature/>
- [4] LM335. Datasheet. [Online]. Available: <https://www.ti.com/document-viewer/LM335/datasheet>
- [5] LDR. Datasheet. [Online]. Available: https://www.nteinc.com/resistor_web/pdf/LDR-Series.pdf